

# Computer & AUTOMATION

Fachmagazin der Fertigungs- und Prozesstechnik

Dr. Carsten Emde, Hans Jürgen Rauscher

## Linux – die Varianten

Linux selbst erstellen oder doch eine Distribution kaufen? Dieser strategischen Entscheidung sollte eine kritische Analyse der Implementierungsvarianten und der eigenen Ressourcen vorausgehen.

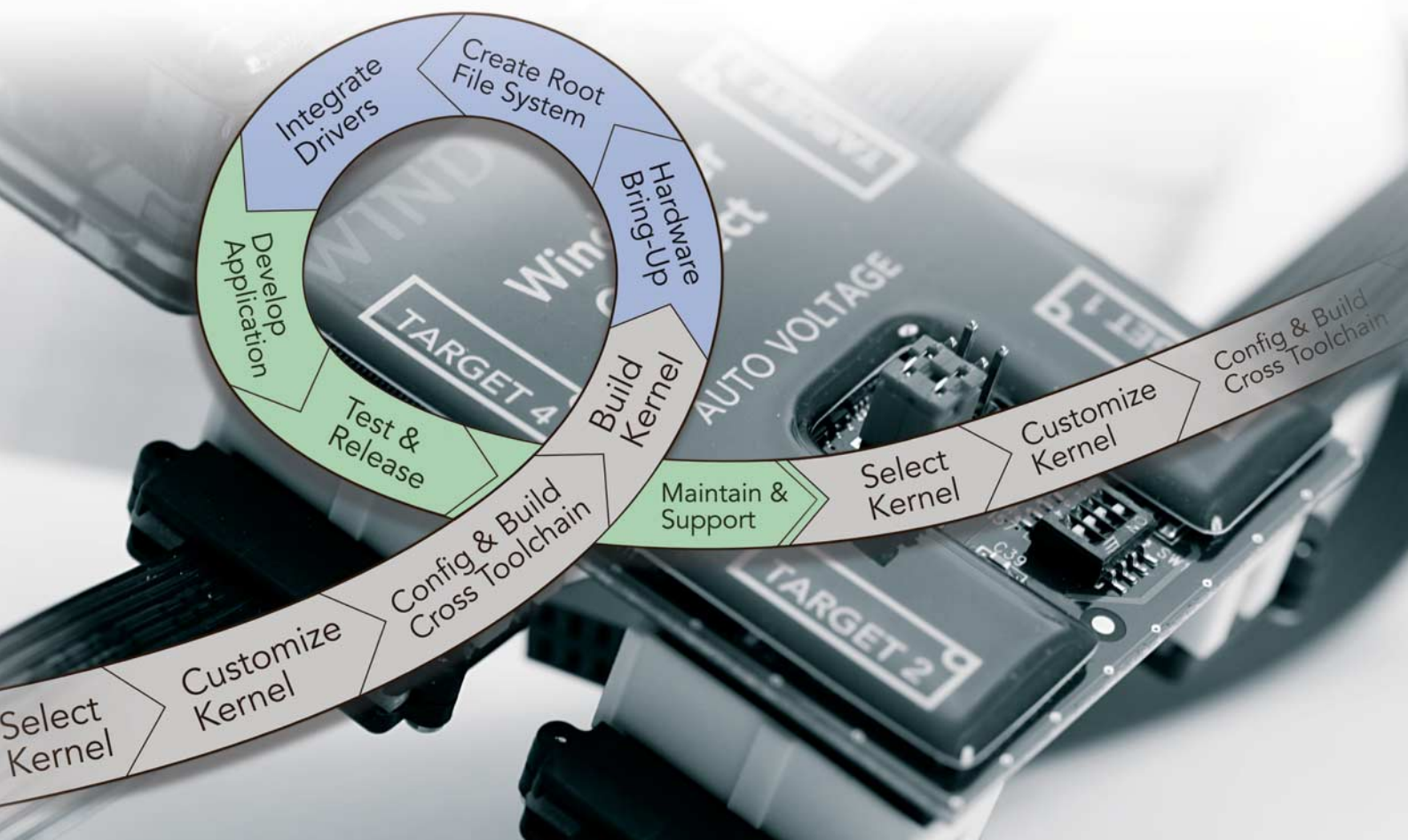
Die Anforderungen an ein Betriebssystem im Industrie- und Automatisierungsbereich sind umfangreich: Echtzeit-Fähigkeit für Steuerungen, umfangreiche Grafikfunktionen für HMI-Terminals und ausgeprägte Kommunikationsfähigkeit mit Feldbussen, Realtime-Ethernet und TCP/IP für die Vernetzung. Dazu kommen Anforderungen wie ein sparsamer

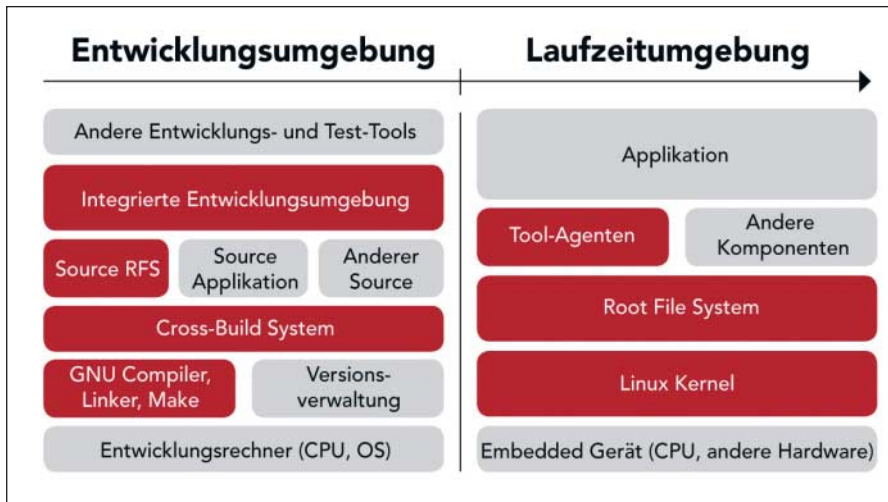
Umgang mit CPU- und Speicherressourcen sowie die Unterstützung verschiedener Hardware-Plattformen wie VMEbus, CompactPCI und in Zukunft voraussichtlich auch MicroTCA. Viele dieser Anforderungen erfüllt Linux inzwischen, die Echtzeit-Fähigkeit des ungepatchten Mainline-Kernels allerdings noch nicht vollständig.

### Echtzeit auf Raten

Eine Möglichkeit, Linux mit Echtzeit-Eigenschaften auszustatten, besteht in der Verwendung von Realtime-Extensions. Deren bekannteste Vertreter sind RT-Linux, RTAI oder Xenomai. Diese Lösungen basieren auf einem echtzeitfähigen Mikrokern, auf dem eine modifizierte Version des Linux-Kernels als Task läuft.

(Bild: Computer & AUTOMATION, Quelle: Wind River)





Die Komponenten (rot unterlegt) einer Linux-Plattform (freie oder kommerzielle Distribution) sollen die Host- wie auch die Target-Seite abdecken.

Dadurch bleibt die Kompatibilität mit Standard-Linux-Applikationen erhalten. Ihr Nachteil ist, dass die notwendigen Modifikationen am Linux-Kernel vermutlich nicht in den Mainline-Kernel aufgenommen werden und dadurch jedes Release des Linux-Kernels immer wieder angepasst werden muss.

Unabhängig von den Realtime-Extensions begannen etwa seit dem Jahre 2000 die Entwicklungen einer anderen Möglichkeit, den Linux-Kernel mit Echtzeit-Fähigkeit auszustatten. Diese Realtime-Preempt-Patches sind bereits zu etwa 60 % im aktuellen Mainline-Kernel 2.6.22

**Ablauf einer Linux-Implementierung: Kommerzielles Linux reduziert den Aufwand beim Projektstart und sorgt so für einen Vorsprung von etwa 12 Wochen gegenüber Linux im Eigenbau.**

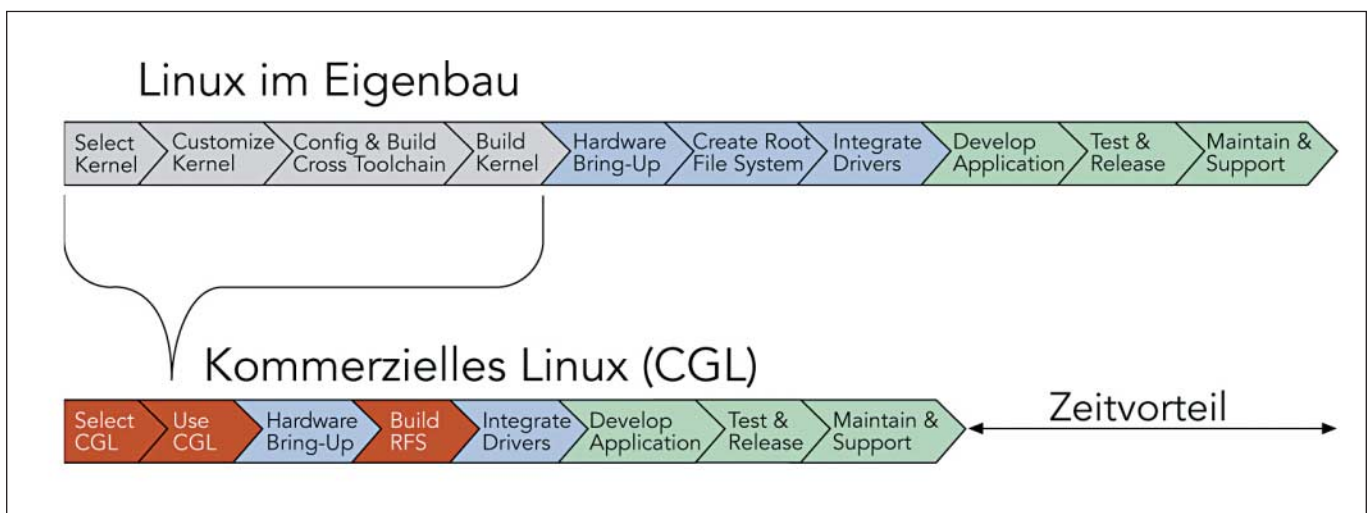


implementiert. Die fehlenden Teile, wie zum Beispiel das "Interrupt-Threading", sollen schrittweise in die nächsten Kernelversionen einfließen – ein Vorgang, der bis Mitte 2008 abgeschlossen sein soll. Bis dahin müssen die jeweils noch nicht im Mainline-Kernel enthaltenen Komponenten der Realtime-Preempt-Patches vor dem Übersetzen des Kernels eingespielt werden.

### Vom Linux-Kernel zum System

Alle Linux-Systeme werden nach dem gleichen Schema erstellt: Am Anfang steht die Auswahl des Linux-Kernel, in der Regel eine möglichst aktuelle Version, die für die vorgesehene CPU-Architektur beziehungsweise für das verwendete Board verfügbar ist. Hinzu kommen die Patches für Funktionserweiterungen wie Echtzeit-Fähigkeit und zur Unterstützung von speziellen Controllern.

Anschließend folgt die Wahl der dazu passenden, bereits compilierten GNU-Toolchain (Programmierwerkzeuge), beispielsweise die GNU Compiler Collection (GCC), GNU Binary Utilities (binutils) und Debugger (GDB) sowie die Standard-C-Bibliothek (glibc) oder eine reduzierte Variante. Im letzten Schritt wird der Kernel übersetzt und mit den benötigten Treibern, Dateisystemen sowie Netzwerk-Protokollen in einem Root-Filesystem zusammengefasst. In der Regel sind für diese Arbeiten mehrere Monate anzusetzen. Da die einzelnen Schritte aufeinander aufbauen, lassen sie sich nur bedingt durch paralleles Arbeiten mehrerer Programmierer beschleunigen. Abhängig vom vorhandenen Know-how, der Unternehmens-Strategie und der verfügbaren Zeit sind allerdings mehrere Implementierungsvarianten möglich:



	Unternehmens-IT	Embedded-Geräte
Fokus	Betrieb von Geschäftsanwendungen	Entwicklung und Vertrieb von Embedded-Geräten
CPU	x86 (Intel, AMD)	X86, PPC, ARM, MIPS, Cell, SuperH usw.
Hauptspeicher	Gigabytes	Megabytes
Speichermedien	Festplatten (Terabytes)	keine, Flash, Festplatten usw.
Linux	Redhat, Novell/SuSE usw.	Wind River, Monta Vista usw.
Kernel	Standard (Distribution)	angepasst (BSP, Patches, Größe usw.)
Toolchain	Nativ (x86 -> x86)	Cross (x86 -> x86, PPC, ARM, MIPS, Cell usw.)
Board-Support-Package	Standard (Distribution)	modifiziert, optimiert, neu entwickelt
Community	Groß (Server, PCs), 85 %	zersplittert (nach CPU & vertikaler Markt), 15 %
Rechtliche Risiken	Gering (Freiheit der Verwendung)	mittel bis hoch (Vertrieb von Geräten, Verpflichtung zur Source Code-Weitergabe)
Anwendung/Fazit	einfach und standardisiert	komplex und gerätespezifisch

◀  
**Linux-Distributionen für die Unternehmens-IT unterscheiden sich grundlegend von Embedded-Distributionen.**

- ▷ Linux im Eigenbau,
- ▷ freie Standard-Distributionen,
- ▷ Linux vom Hardware-Anbieter oder
- ▷ kommerzielles Linux.

Der Vorteil von *Linux im Eigenbau* liegt in der vollständigen Beherrschung des Systems. Hier sind sämtliche Schritte vom Kernel bis zur Applikation selbst zu erledigen. Als problematisch zeigt sich der relativ große Zeitaufwand. Schneller geht es mit der Unterstützung eines Linux-Service-Provider, Linux-Consultants oder Open-Source-Software-Organisationen wie das Open Source Automation Development Lab (OSADL). Einige Linux-Service-Provider bieten eigene Toolchains an, zum Beispiel ELDK (Denx) und OSELAS (Penguintronix). Auch ist zu beachten ist, dass die Anpassung und kontinuierliche Wartung, beispielsweise für das Einpflegen von Sicherheits-Patches, langfristig Entwicklungs-Ressourcen binden, die bei der Applikations-Entwicklung eventuell fehlen.

Im Gegensatz dazu bieten die *freien, das heißt kostenlosen Standard-Distributionen* einen komfortablen Einstieg, da die grundlegenden Arbeiten bereits erledigt sind, beispielsweise die Auswahl, Integration und Zusammenstellung der Software-Pakete. Die Anpassung freier Distributionen an spezielle Computer-Hardware kann aber durchaus sehr aufwendig sein – besonders bei einer geringen Prozessor-Performance und Speicherkapazität der Zielplattform. Daher liegt der Einsatzschwerpunkt von Distributionen wie Debian, Fedora, Open-

SuSE und die auf Sourcen von Redhat Enterprise Linux basierenden Distributionen wie CentOS vor allem bei Standard-Hardware mit x86-CPU's; zunehmend unterstützen diese Distributionen auch PowerPC-Prozessoren. Für Fedora wird darüber hinaus die Bereitstellung einer Version für ARM-Prozessoren diskutiert.

Viele Hardware-Anbieter liefern ihre Referenz- und Produkt-Boards mit eigenen Linux-Versionen aus. Die Bandbreite reicht vom binären, auf das Board maßgeschneiderten Linux-Kernel bis zur umfangreichen Distribution mit Bootloader, Toolchain und Cross-Build-Unterstützung. Diese „Starter-Kits“ sollen die Lauffähigkeit der Hardware demonstrieren und einen schnellen Projektstart ermöglichen. Häufig basieren diese Starter-Kits jedoch auf älteren Kernel-Versionen, die nicht regelmäßig aktualisiert werden. Wenn die mitgelieferten Sourcen dann unverständlich und unübersichtlich sind, kann es schwierig bis unmöglich sein, ein Starter-Kit an eine aktuelle Kernel-Version anzupassen.

Es ist aber durchaus möglich, ein vom Hardware-Anbieter im Quellcode bereitgestelltes Board-Support-Package zu verwenden. Deren Board-spezifische Patches lassen sich bei Bedarf in andere Linux-Kernel implementieren.

Bei den *kommerziellen Linux-Distributionen* gibt es zwei Gruppen: Distributionen von Redhat oder Novell/SuSE haben ihre Domäne im Bereich der Unternehmens-IT als Betriebssystem für Server und Arbeitsplatzrechner. Die Distributio-

nen von Firmen wie Wind River, Monta Vista und anderen sind dagegen auf die speziellen Anforderungen von Embedded-Systemen spezialisiert. Dazu zählen die Unterstützung vieler CPU-Architekturen, die Skalierbarkeit hinsichtlich Speicherbedarf und die Fernwartbarkeit. Beiden Distributionstypen gleich ist die Zeitersparnis beim Einsatz. Nach der Installation und Konfiguration können diese mit kurzer Vorlaufzeit genutzt werden. Vergleichbar sind auch die Maintenance- und Support-Angebote: Die Hersteller aktualisieren ihre Distributionen und bringen mehrmals pro Jahr neue Versionen heraus – natürlich im Rahmen kostenpflichtiger Lizenzvereinbarungen. Einige Anbieter offerieren weitere Dienstleistungen, von der Beratung bis hin zur Entwicklung und Integration spezieller Software oder etwa umfangreiche Entwicklungs- und Testunterstützung wie der Eclipse-basierenden Workbench Development Suite von Wind River.

Bei der Auswahl einer kommerziellen Embedded-Distribution und auch bei der Entscheidung für einen Linux-Service-Provider gilt es einige Punkte zu beachten:

- ▷ Stellt der Anbieter die notwendigen Entwicklungs-Tools wie Compiler, Debugger, aber auch weitergehende Testwerkzeuge wie Diagnosefunktionen zum Aufspüren von Fehlern im laufenden Betrieb eines Geräts bereit?
- ▷ Wird der komplette Source-Code mitgeliefert und lässt sich daraus der eben-



## Checkliste

### Die Linux-Varianten

Die Implementierungs-Strategie bei Linux hängt von vielen Faktoren ab:

#### Ressourcen & Know-how

- ▷ Wie ist die Zeitverteilung der Wartung bestehender Produkte versus Entwicklung neuer Produkte?
- ▷ Wieviele Entwickler mit Linux-Erfahrung sind im Haus verfügbar?
- ▷ Wieviele können sich um die Entwicklung und Pflege von Linux und Linux-Tools kümmern?

#### Dienstleistungen

- ▷ Wer beantwortet technischen Fragen (Support)?
- ▷ Wer pflegt den eigenen Linux-Code (Maintenance)?
- ▷ Wer berät (Consulting)?
- ▷ Wer portiert den Linux-Code eventuell auf ein anderes Board oder eine andere CPU-Architektur?
- ▷ Wer hilft bei der Implementierung (Services)?

#### Entwicklungsumgebung

- ▷ Welche Tools nutzen?
- ▷ Gibt es Vorgaben der Unternehmens-IT hinsichtlich Entwicklungsrechner, Versionsverwaltung?
- ▷ Wie ist die Integration von Tools vom Hardware Bring-up bis zum Systemtest?

#### Laufzeitumgebung

- ▷ Welche CPU und welches Board wird eingesetzt?
- ▷ Welche Kernel-Version?
- ▷ Welche Treiber, Kernel-Module und Pakete sind notwendig?
- ▷ Wo muss das Root-Filesystem liegen?
- ▷ Wird die Applikation portiert oder neu erstellt?
- ▷ Bestehen Sicherheitsanforderungen (SELinux)?
- ▷ Welche Echtzeit-Eigenschaften sind notwendig?

anschlagten Projektzeit ein funktionsfähiges Gerät mit größerem Leistungsumfang und möglichst vor dem Mitbewerber auf den Markt zu bringen. Die Wahl der richtigen Strategie kann dazu einen wesentlichen Beitrag leisten. Denn oft wird der tatsächliche Aufwand für ein Linux im Eigenbau unterschätzt. *sk*

#### Nähere Informationen:

The Real Costs of Roll-Your-Own-Linux:  
<http://www.windriver.com/whitepapers/Realtime-Preemption-Patches>:  
[www.osadl.org/Realtime-Preempt-Kernel/kernel-rt.0.html](http://www.osadl.org/Realtime-Preempt-Kernel/kernel-rt.0.html)

#### Artikel-Download:

[www.elektroniknet.de/automation](http://www.elektroniknet.de/automation)



**Dr. Carsten Emde**

ist Consultant für Embedded-Linux und Geschäftsführer des Open Source Automation Development Lab (OSADL) in Schopfloch.



**Hans Jürgen Rauscher**

ist System Architect Networking bei der Wind River GmbH in Ismaning.

falls ausgelieferte Binärcode (pre-build binaries) exakt nachbauen?

- ▷ Existiert eine verlässliche Produkt-Roadmap, die Planungssicherheit für eigene Projekte gibt?
- ▷ Ist der technische Support in der gleichen Zeit- und Sprachzone ansässig?
- ▷ Gibt es bei Bedarf Training-, Beratung oder Entwicklungsunterstützung?
- ▷ Hat der Anbieter Branchenkenntnis?

▷ Passt das Lizenzmodell zum Projekt und der Firmengröße?

- ▷ Gibt es ein Partner-Programm, das Software- und Hardware-Unterstützung für die Embedded-Linux-Distribution anbietet?

#### Keine Zeit zum Spielen

Abgesehen von ersten kleinen Testprojekten, herrscht bei jedem Projekt Erfolgs- und Zeitdruck. Es gilt innerhalb der ver-

Open  
Source  
Automation  
Development  
Lab eG



Ohne  
Sorge  
Automatisieren  
Dank  
Linux

"Das Ganze ist mehr als die Summe seiner Teile"

Aristoteles (384 - 322 v. Chr.)

"Was dem einzelnen nicht möglich ist, das vermögen viele"

Friedrich Wilhelm Raiffeisen (1818 - 1888)

"Durch Open Source wird jede individuelle Verbesserung zu einem Gewinn für alle"

Linus Torvalds (\*1969)

OSADL eG, Homagstr. 3-5, 72296 Schopfloch, Tel 07443/13 3073, Fax 07443/13 8 3073

Email [info@osadl.org](mailto:info@osadl.org), Web <http://www.osadl.org>